Service vs Protection: A Bayesian Learning Approach for Trust Provisioning in Edge of Things Environment

Parminder Singh, *Member, IEEE,* Avinash Kaur, Ranbir Singh Batth, *Member, IEEE*, Gagangeet Singh Aujla, *Senior Member, IEEE*, and Mehedi Masud, *Senior Member, IEEE*

Abstract—Edge of Things (EoT) technology enables end-users participation with smart-sensors and mobile devices (such as smartphones, wearable devices) to the smart devices across the smart city. Trust management is the main challenge in EoT infrastructure to consider the trusted participants. The Quality of Service (QoS) is highly affected by malicious users with fake or altered data. In this paper, a Robust Trust Management (RTM) scheme is designed based on Bayesian learning and collaboration filtering. The proposed RTM model is regularly updated after a specific interval with the significant decay value to the current calculated scores to update the behavior changes quickly. The dynamic characteristics of edge nodes are analyzed with the new probability score mechanism from recent services' behavior. The performance of the proposed trust management scheme is evaluated in a simulated environment. The percentage of collaboration devices are tuned as 10%, 50% and 100%. The maximum accuracy of 99.8% is achieved from the proposed RTM scheme. The experimental results demonstrate that the RTM scheme shows better performance than the existing techniques in filtering malicious behavior and accuracy.

Index Terms—Edge of Things (EoT), Trust management, Machine Learning, Smart city, Smart Sensors, Malicious attack.

1 Introduction

With the evolution of the Internet of Things (IoT) [1], smart devices have transformed life and emerged as an excellent benefit for the world. The proliferation of IoT and its success provides a new horizon of computing called the Edge computing [2], where technologies perform computation at the edge of the network. Edge computing has solved issues of response time, bandwidth cost saving, data safety, and privacy [3] [4]. EoT allows on-device computing and analytics. In edge computing, devices manage decisions themselves in a real-time environment, which requires instantaneous responses for technologies like autonomous vehicles, where the human-like reactions are integral to achieve higher levels of comfort and safety. EoT also has drawn significant attention to future intelligent transportation systems [5] especially in smart cities data processing near to the user location to maximize the speed and to minimize the latency. Moreover, EoT has become popular in the industry, and academic research with 5G communication technology [6].

Edge computing plays a vital role in smart city development [7] [8] because it relies highly on sensors for decision-making devices, which have become more pervasive and integral to the smart city ecosystem [9], [10]. Advanced technologies like cloud computing [11] and IoT [12] integrate numerous embedded devices that generate a tremendous volume of data that can be leveraged primarily for health, safety, disaster prevention, and infotainment services [13]. In this paper, a framework is designed for the Edge-of-things based smart-city Ecosystem.

In the smart city ecosystem [14], [15], many intelligent edge devices interact in different areas of the city, like highways, buildings, and stadiums. Here, users connect with their smart devices (e.g., mobile phones, smart wearable devices) that submit data to edge service providers for processing [16]. The major challenge of an edge service provider is resource limitation [17]. For instance, if an edge service provider receives a large volume of data for processing, which can lead to delay and service latency. And to solve this issue, the smart terminal shifts the load to the other smart devices with idle resources. Hence, the main obstacle is to choose the reliable and the most trusted device as all the available intelligent non-trusted devices in the network. The selection of a non-trusted device may cause network damage or affect the QoS [18]. Various kinds of security risks and different malicious attacks are also a challenge in the smart city ecosystem. Trustworthiness is a vital concern because there is a possibility of misleading by the malicious user through the fake edge devices. Security is a vital cornerstone for EoT as the integrity of data trust in services for delivering data is crucial [19] [20].

Therefore, trust management is a crucial factor in ensuring the quality of assistance to the end-users. However, most trust models typically follow generic steps such as collecting behavioral information, rating and ranking entities, selecting entities, transactions, and rewarding. The key challenge of the EoT ecosystem in smart cities is to select the trusted edge nodes, as many nodes in the EoT environment can be

Corresponding Author: Ranbir Singh Batth
P. Singh, A. Kaur, and R. S. Batth are with the School of Computer Science and Engineering, Lovely Professional University, India, 144411. email: parminder.16479@lpu.co.in, avinash.14557@lpu.co.in, ranbir.21123@lpu.co.in
G. S. Aujla is with the Department of Computer Science, Durham University, United Kingdom. email: gagi_aujla82@yahoo.com, gagangeet.s.aujla@durham.ac.uk
Mehedi Masud is with the Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia

untrusted. These untrusted edge nodes carry the potential to damage the entire services or network maliciously. Smart devices have the potential to improve our quality of life and make our lives more comfortable. However, it involves various malicious attacks and security risks in the edge-of-things system in the smart city. Trust management is used by a service provider of edge-of-things to assure the high performance of intelligent devices' behavior, leading to improved user satisfaction.

The value of trust between edge service providers and smart devices is ignored in trust value calculations. The previously proposed models ignored the independent choice of the participant. This paper introduces a selective recommendation method for smart devices based on Bayesian learning and collaboration filtering to find a solution to some of the trusted participants' problems. The proposed method adds significant decay value to the trust score for quick updates with the change in edge nodes' behavior. The proposed method outperforms the existing trust model in most cases of edge computing. This research also introduces the probability score mechanism for studying the proposed scheme of trust management's stability and validity. The stability and validity of the trust management scheme are proved with box and whisker plots. The trust management scheme's effectiveness is verified by air quality monitoring, personal health care monitoring system, and system of analysis in a smart city.

The main contributions of this research are:

1) Evaluated the trust scores for scalable smart cities to form multiple edge-centers.
2) Proposed a Robust Trust Management scheme using collaboration filtering and significant decay for edge-nodes trustworthiness and faction-nodes.
3) Developed an algorithm for the recommender system for appropriate node selection with the awareness of Quality of Service (QoS) and Quality of Protection (QoP).
4) Compared the performance of the proposed RTM with the existing trust models.

The organization of the paper: Section 2 discusses related work. Section 3 presents the Edge of Things (EoT) model for smart cities. Section 4 discusses the proposed Robust Trust Management scheme for the smart city ecosystem under EoT infrastructure. The performance of the proposed RTM model is demonstrated in Section 5. Finally, section 6 concludes the paper by highlighting some future work.

2 Related Work

Su *et al.* [21] introduced a trust management system that is resilient to malicious attacks. It has three unique features. First, it encapsulates multi-scale quality-sensitive feedback and integrates user behavior variances into a local trust algorithm. Second, service trust tests the similarity of two users' input actions and aggregates the local trust value into a global trust algorithm by using similarity scores in pairs to weigh the contribution of local trust values against a participant's global trust. Finally, weighted trust propagation similarity feedback is used to improve global trust computing's robustness against malicious feedback. Fan *et al.* [22] presented dependable trust management aspects in a group. The authors explained the continuous growth of open systems when multiple entities try to interact with each other without any prior information regarding the system, where entities have no information about the system. The authors made three contributions: finding out vulnerabilities in a model, increasing factors, and finally checking the effectiveness of an open system. The authors realized that group trust outperforms another trust model.

Hui *et al.* [23] proposed a mechanism for security in mobile edge computing, which provides the ability of distribution on the network edge. The authors proposed a resource allocation mechanism using a stability theory to overcome the Mobile-Edge Computing Intrusion detection systems' efficient resource allocation problem. The authors studied security issues in the MEC network and found that the efficient allocation of resources in the MEC environment is a very critical and challenging task. To tackle these challenges, the authors proposed a new mechanism evaluating the differential equation model. Din *et al.* [24] explained the internet of things' trust management techniques. The author surveyed the IoT future when different networking devices are connected to form a network using an IP address. When this connected network generates data again, the user's trust dependability plays a vital role in sharing data.

Li *et al.* [25] proposed a personal trust reputation management model which overcomes the problem faced by Eigen Trust when peers and spies are different. In this model, any peer can propagate based on the pre-trusted peers. Peers interact in the network to find the pre-trusted peers. The pre-trust matrix is updated automatically, and a white list is generated for the dynamic optimization of the pre-trust matrix. This white list improved the performance of the model as it included all peers with the pre-trust set. Yuan *et al.* [26] proposed a trust mechanism for the IoT-enabled intelligent edge devices, which is reliable and lightweight as compared to others. The mechanism relies on the multi-source feedback information. Here, trust calculation is fully completed by the broker layer and device layer. The authors selected the lightweight trust evaluating mechanism as it is suited for large-scale IoT edge-enabled computing.

Gessner *et al.* [19] focused on guaranteeing data privacy and confidentiality to the users by enhancing trust in the services. The authors introduced resolution infrastructure, which worked on IoT security by enhancing trust's functional security components. Three essential functions in the study are trust and reputation management (TRA), key exchange and management (KEM), and identity management (IM). These components provide a higher level of security by providing secure communication. Chen *et al.* [31] designed a scalable and adaptive trust management protocol for service-oriented architecture based on IoT devices. The authors applied a simulation-based filtering technique to collect Trust feedback for IoT nodes having similar social interests. This adaptive filtering technique further adjusted itself to identify and combine direct and indirect trust. In terms of accuracy, resilience, and trust convergence, the proposed protocol gave better results when compared to PeerTrust and EignTrust. Further, the authors introduced a storage management method for IoT devices to promote the scalability factor. Yan *et al.* [32] provided a comprehensive

TABLE 1: Comparison of various trust management schemes

Ref.	System Model	Information gathering	Trust computation	Trust propagation	Trust update	Performance Metrics	Experiment
[21]	Service Provision Networks	Direct	Multi-scale rating	Normalization	Time-driven	Failed services	Simulation: file sharing service network
[22]	Peer-to-Peer	Direct	Similarity based	Susceptible Infected Recovered (SIR)	Normalization and feedback	Time overhead, Trust score	Simulation with synthetic and real datasets
[25]	Peer-to-Peer	Direct	Local and global pre-trust matrix	White list with good services rate	Time-driven	Inauthentic downloads	TR/RM simulation
[26]	IoT Edge computing	Indirect	Objective information entropy theory	Direct trust + B2D trust	Time-driven	Propagation of malicious nodes	NetLogo event simulator
[27]	Crowd Source IoT Services	Direct	Neural Network + Adam's optimizer	Highly trusted, Neutral trusted, Lowly trusted, and Not trusted	Training based	Accuracy, recall, and precision	Amazon Mechanical Turk (MTruk)
[28]	IoT Edge Computing	Indirect	Evolutionary game theory	Lyapunov theory, black and white list	Time-driven	Success ration on attack models	Theoretical and simulation
[29]	Social Internet of Things	Indirect	Hellinger distance	Matrix factorization	Time-driven	MSE, RMSE, Trustworthiness	Simulation
[30]	Vehicular Social Networking	Indirect	Grounded theory	Functional algorithms, intuition-based methods	Time-driven	Accuracy of crowdsourced data analysis	Theoretical and simulation
Our RTM	Edge of Things (EoT)	Direct and Indirect	Bayesian learning and collaboration filtering	Faction edge nodes recommendation	Time-driven + Decay value	Performance on attack models	Simulation

survey on trust properties essential for achieving high trust management. They made five classifications of these trust properties. The study proposed ten objectives named: Trust relationship and decision, Data perception trust, Privacy preservation, Data fusion and mining trust, Data transmission and communication trust, Quality of IoT services, System security and robustness, Generality, Human-computer trust interaction, and Identity trust to get high trust services. Moreover, based on general IoT architecture.

Wang *et al.* [28] proposed a trust management scheme for a smart city equipped with an IoT edge computing system. The evolutionary game theory strengthens trust management stability and validity. Furthermore, the proposed system escalates the cooperation of IoT devices in edge computing. Bahutair *et al.* [27] proposed a multi-perspective trust model that takes into account the inherent characteristics of crowd-sourced IoT services. Each perspective has a set of characteristics that contribute to the perspective's ability to influence trust. Aalibagi *et al.* [29] suggested a mechanism that is resilient to various forms of network attacks. The proposed technique can reliably identify the most suitable and secure service provider. Liu *et al.* [30] addressed the problems surrounding trust management in international crowd-sourcing initiatives centered on large volumes of untrustworthy data. This approach is intended to minimize the number of factors influencing the quality of crowd-sourced data analysis. The detailed comparison of various studies is described in Table 1.

3 System Model

This section presents an EoT framework for multiple edge locations with multiple devices. The framework considers the scalable EoT ecosystem with enabled mobility. Figure 1 shows the trust management scheme designed for the EoT framework. Different attack models are evaluated varying the number of malicious nodes in this framework.

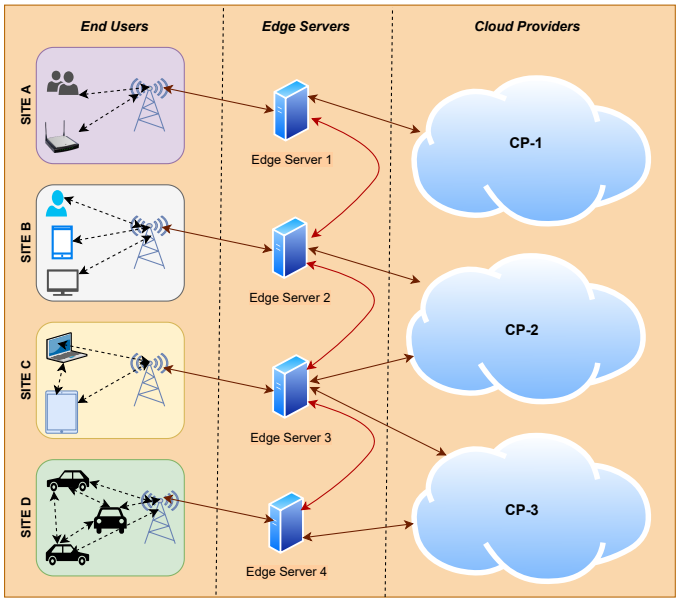


Fig. 1: Edge of Things (EoT) framework for smart cities

3.1 Edge of Things (EOT) Framework for Smart Cities

Figure 1 shows the framework of edge computing of the smart city based on multi-edge centers and multi-terminal.

The devices are deployed logically or physically near the users or end devices. As the devices possess lower latency, lead to a reduction of the response time of the job, hence leads to the improvement in the Quality of Experience (QoE) of the user. It reduces cloud computing dependency on network bandwidth, minimizes denial of service attacks, and improves service availability. The edge computing framework with multi-edge and multi-terminal centers based on cloud computing platforms contains three parts: remote cloud service providers, edge service providers, and end-users, as shown in Figure 1.

3.1.1 End Users

The device uses wireless or wired networks for accessing edge service providers. It leads to perform storage or computing capabilities being defined for the end-users. The edge service provider closest to the access is chosen by the end-user, thus reducing submitted requests' processing time. When the end-user completes the job, service feedback is provided to the edge service providers. Before starting the new task, the request is sent to the edge service provider to obtain credibility with a collaborator.

3.1.2 Edge Servers

In the same area, Edge service providers consist of related devices and servers. The service requests are processed and released near providers of edge services from several users. It provides storage, computing, and several services to end-users flexibly and quickly. End-users monitor the service behavior of devices. Also, evaluation feedback from users and devices is aggregated. In the end, results are sent to cloud data centers. In the real environment of edge computing, malicious or unreliable devices may lead to wrong evaluation feedback results. The traditional evaluation feedback method is extended for the reduction of risks in the edge-of-things. Hence, it improves the reliability of the system.

3.1.3 Cloud Service Providers

The coordination of edge service providers completes the cloud computing services required by the end-users. The services and resources of edge service providers are monitored and managed by remote cloud service providers. In a real-time environment, resource usage and operation status are controlled by a remote cloud service provider. The resources and services of each edge service provider are collected, then scheduling of services and resources is performed by the remote cloud service provider.

4 Robust Trust Management Scheme

This section discusses the proposed RTM scheme. In RTM, the Edge nodes collaborate with others. There are two types of Edge nodes: Malicious and Trusted. Malicious edge nodes are under attack and cause a security breach in the network. The trusted nodes provide the desired services with full capacity. Malicious nodes can attack the trusted nodes. The proposed trust management scheme helps find the malicious nodes and avoid mechanisms to save the trusted nodes.

4.1 Edge Nodes Trust Association

Quality of Protection (QoP) and Quality of Service (QoS) parameters are used to calculate the satisfaction level of a requester user u_r for the services given by a provider user u_p. Trust T evaluation is performed for each interaction as $T(e_a, e_b)$ from edge node e_a to edge node e_b. If u_r is satisfied, then the score would be 1 else the score would be -1. The association score is stored at each edge node for every deal to forecast the trust for the next interaction. This association between the edge nodes is known as direct experience because the trust is evaluated based on the node's own experience. Trust Level (TL) is calculated based on these scores. The TL score becomes high for the nodes having a positive trust history. The trustworthiness of each node can be evaluated from the historical log. The overall trust score of $\hat{T}$ between two nodes is represented with $\hat{T}_{a,b}$. It is essential for each edge node to store the trust score $\hat{T}$ for direct interaction.

The Bayesian Network is used for calculating the trust score for direct interaction. Bayesian gives an authentic performance for the modeling of peer-to-peer network [31]. Each interaction represents a satisfactory or unsatisfactory experience and is denoted by the test score is either 1 or -1. The trust of receiver and provider edge nodes $\hat{T}_{a,b}$ is calculated as per Equation 1.

$$\hat{T}_{a,b} = \frac{\alpha_{T(e_a,e_b)}}{\alpha_{T(e_a,e_b)} + \beta_{T(e_a,e_b)}} \tag{1}$$

In the above equation, trust score $\hat{T}_{a,b}$ is calculated between the two nodes a and b. The Beta distribution parameters are represented with $\alpha_{T(e_a,e_b)}$ and $\beta_{T(e_a,e_b)}$. A satisfactory experience is stored in the $\alpha_{T(e_a,e_b)}$, where as $\beta_{T(e_a,e_b)}$ applied to log the unsatisfactory experience. After certain time $\triangle t$, the calculated value of $\alpha_{T(e_a,e_b)}$ and $\beta_{T(e_a,e_b)}$ are updated as Eq. 2 and Eq. 3.

$$\hat{\alpha}_{T(e_a,e_b)} = T(e_a, e_b) + (e^{d\triangle t} \times \alpha_{T(e_a,e_b)}) \tag{2}$$

$$\hat{\beta}_{T(e_a,e_b)} = T(e_a, e_b) + (e^{d\triangle t} \times \beta_{T(e_a,e_b)}) \tag{3}$$

The old trust scores are represented with $\alpha_{T(e_a,e_b)}$ and $\beta_{T(e_a,e_b)}$ and new score are refer with $\hat{\alpha}_{T(e_a,e_b)}$ and $\hat{\beta}T(e_a, e_b)$. The decay factor d represents the trust decay for an on-going clock. The decay is updated after a specific interval $\triangle t$. The decay value help us to decline the trust score if two nodes are not interacted for a period of time. To make the technique robust, we consider the scalable EoT infrastructure. The higher weight to the recent scores is given through the decay mechanism. The small value of decay is added to the overall trust score over time. The interaction of two edge nodes $e_a \rightarrow e_b$, where a node requests the services from edge node b. The mapping of the trust score is performed with the following parameters.

- $e_a \rightarrow e_b$ - interaction from edge node a to edge node b
- N - number of interactions
- $\alpha_{T(u_r,u_p)}, \beta_{T(u_r,u_p)}$ - old trust scores
- $\hat{\alpha}_{T(u_r,u_p)}, \hat{\beta}_{T(u_r,u_p)}$ - new trust scores
- TL - Trust Level

Most researchers consider the initial value of trust for α and β either 0, 1, or null, as there is no prior knowledge of the interaction behavior for edge nodes [31]. This information is obtained from the neighboring nodes having the same set demand for direct interaction services in the proposed RTM model. Suppose no prior knowledge is available for the direct experience of interaction. In that case, the central edge server or cloud server is used to store the TL for every edge node in the smart city. Then, distributed Collaborating Filtering (CF) [31] is used to calculate the trust score for the edge nodes. CF works in two steps:

1) Trusted Edge Nodes Recommendation: Once the trust score is calculated, the recommender system is designed for the TL based on edge nodes' interaction. If the direct trust score is satisfactory, then there is no requirement to calculate the Protection Requirement (PR). PR is the required security factor for healthy communication in the EoT environment. The QoP is measured using PR; the higher the PR, the higher the QoP.
2) Faction Edge Nodes Recommendation: The recommendation is taken from the services provided to another edge node on a different smart city as a faction edge node. This value can be obtained through the edge servers as shown in the smart city framework as shown in Figure 1. It is used to get the desired PR and Secure Service Level Agreement (SSLA). The trust score must be matched with the desired edge node.

Trusted edge nodes and faction edge nodes are used in the proposed RTM scheme. The neighboring edge nodes and trustor edge node relationships are analyzed in the first phase to refrain from the invader edge nodes. The recommendation is used only to consider if their TL score is satisfactory. If the faction serves as a trustor, the recommendation is based on SSLA fulfillment.The total recommendation is calculated based on Eq. 4.

$$RM(a,b) = \sum_{w \in R} RM_{e_a,e_b} \times sv_w \tag{4}$$

The significance value of recommendation is taken from trusted edge nodes sv_u and faction edge nodes sv_f. The recommendation from a to b edge node is described with RM_{e_a,e_b}. Once the response is received, the edge node can recommend with a significant trust value 1 or -1. Therefore, the overall test score is calculated as per Eq. 5.

$$\hat{T}_{a,b} = \frac{RM_{e_a,e_b}}{\sum_{i=0}^{N.RM} RM_{e_a,e_b}(a,b)} \tag{5}$$

The detailed process of recommendation is explained by Algorithm 1. Consider a situation where edge node e_a wants the services from the edge node e_b. In the first step, as per Algorithm 1, the edge node e_a asks the edge nodes for the recommendation. The list of neighboring edge nodes E_N is obtained and entered into the Edge List E_L(line 1).The Edge List has been sorted in descending order as per the Trust Level TL (line no. 2). Each node from the Edge List is picked and calculated as the trust score and stored as Trust Recommendation TR. If the TR is less than 0, delete that Edge Node from the Edge List. Otherwise, calculate the recommendation of faction nodes and update the recommendation in Edge List (line no. 3-11). The final Edge List has been returned (line no. 12). For example, the 3 neighbouring nodes to e_a are e_x, e_y, e_z. The request for recommendation is requested from the trusted edge nodes. The request for recommendation is sent with a function call F_{RM}(e_b,SSLA). The trustworthiness is calculated for e_b. The second part of this is the secure service level agreement (SSLA). This value is set per the requirement of services by edge node e_a defined with PR. As per SSLA requirement of e_a, the edge nodes e_x, e_y, e_z calculate the trustworthiness as per the proposed RTM trust management scheme. The untrusted nodes are deleted from the edge nodes linked list. The overall trust score is calculated as per Eq. 5 and final recommendation is sent back based on Eq. 4.

Algorithm 1 Edge Nodes Recommendation Scheme

Input: SSLA, Edge Node A (e_a), Edge Node B (e_b)
Arguments: Recommendation (RM), Edge List E_L, Trust Score $\hat{T}_{a,b}$
Output: Trust Level (TL) from e_a to e_b

1: $E_L \leftarrow List[E_N]$ ▷ get the list of edge nodes
2: $E_L \leftarrow$ sortdescending(E_L,TL) ▷ list descending order as per trust level
3: **for** E_i in E_L **do**
4: $TR \leftarrow$ TrustScore(E_a,E_b) ▷ trust recommendation (TR)
5: **if** $TR < 0$ **then**
6: E_L.Delete(E_i) ▷ delete the Edge node from linked list
7: **else**
8: $E_i \leftarrow F_{RM}$(e_b,SSLA) ▷ recommendation of faction edge node
9: $E_L \leftarrow F_i(RM)$ ▷ recommendation update
10: **end if**
11: **end for**
12: return E_L

4.2 Dynamic Characteristics of Edge Nodes

The dynamic change in the characteristics of the edge nodes must be taken into consideration. For example, once the node gets a high trust score, it may provide malicious or invalid services. In other situations, the nodes with lower trust scores may provide valid and quality services to escalate the trust score. The probability mechanism is introduced to update the dynamic behavior of the malicious services. The probability score of PS is calculated that influences the trust score given by faction nodes. The probability scores of services provided by edge node b to edge node a can be calculated as per Eq. 6.

$$PS_{(a,b)} = \frac{invalidServices_{(a,b)}}{invalidServices_{(a,b)} + validServices_{(a,b)}} \tag{6}$$

The high PS leads to the most malicious services provided by edge node b. In this case, even the faction nodes provide a high trust score, but the edge node b must be removed from the selected edge node list on an individual basis.

The edge node list E_L is further reduced based on the probability score. The detailed work of the probability-based edge node list update process is described in Algorithm 2.

Algorithm 2 Edge node list updation with probability score

Input: Edge List E_L, $Services_{(a,b)}$
Arguments: Edge List E_L, Trust Recommendation (TR)
Output: Edge List E_L

1: $E_L \leftarrow List[E_N]$ ▷ get the list of edge nodes
2: $E_L \leftarrow$ sortDescending(E_L,TR) ▷ list descending order as per trust recommendation
3: **for** E_i in E_L **do**
4: // Calculate the PS as per Eq. 6
5: $invalidServices_{(a,b)} \leftarrow 0$
6: $validServices_{(a,b)} \leftarrow 0$
7: **for** $c = 1$ to N **do**
8: **if** $Service^{c}_{(a,b)} == malicious$ **then**
9: $invalidServices_{(a,b)} \leftarrow invalidServices_{(a,b)} + 1$
10: **else**
11: $validServices_{(a,b)} \leftarrow validServices_{(a,b)} + 1$
12: **end if**
13: **end for**
14: $PS_{(a,b)} = \frac{invalidServices_{(a,b)}}{invalidServices_{(a,b)} + validServices_{(a,b)}}$
15: $TR \leftarrow TR - PS$ ▷ trust recommendation (TR)
16: **if** $TR < 0$ **then**
17: E_L.Delete(E_i) ▷ delete the Edge node from linked list
18: **end if**
19: **end for**
20: return E_L

The Edge list is obtained from the Algorithm 1 is an input to the Algorithm 2. The recent N services are considered for the probability calculation. The user can decide the value of N in order to increase the speed of the process. The initial value of valid and invalid services is 0 (line no. 5-6). After that, the services provided in the history are analyzed in terms of malicious or valid services. The counters are incremented according to conditional statements (line no. 8-12). Further, the probability score of PS calculated from the counters ($validServices$ and $invalidServices$) as per (line no. 14). The trust recommendation is the trust score for the services from E_a to E_b. The probability scores PS are subtracted from the trust recommendation TR (line 15). The larger value of PS has a high impact on the value of TR. The high value of PS is the significance of higher malicious services in the trusted edge nodes' recent provided services. If the TR is less than 0, then that edge node has been removed from the Edge list (line no. 16-17). The final Edge list E_L returned from the algorithm.

5 Experimental Evaluation

5.1 Experiment Setup

In the experiment, the number of end-users are 500. The trusted users are preset to 5. The assumption in this research is that end users posses same recognition υ_i and same tolerance γ_i. In [33] [34] [35], $\gamma_i = \upsilon_i = 10$ have used commonly. In this system, the services are considered as 1000. In the beginning, there are 10 services as an average for end users. Initially, for maintaining the trust relationships between the users, 1000 rounds of warm-up interactions have introduced. The Percentage Collaborative Devices (PCD) is tuned to 10%, 50%, and 100% for the experiment purpose.

This indicates the interaction of smart devices in the smart city is low, medium, and high. The number of service requests by the ordinary user is N_{ou} and the number of valid services is represented by N_{vs}. The Effective Service Rate (ESR) is measured as $ESR = N_{vs}/N_{ou}$. The attack models are represented as *scenario-1* (S1), *scenario-2* (S2) and *scenario-3* (S3). Because there is a resemblance between bad mouthing and good mouthing attacks, they are classified as $s1$. The ballot-stuffing attacks are put at $S2$, and $S3$ refers to the selective-behavior attacks similar to reliable IoT edge computing [28] trust model.

For completion of tasks, an edge-of-things scenario is considered. In the trust management domain, simulations are performed on the basis of synthetic data in [31], [36], [37], [38], [39]. The performance of the proposed model is evaluated with TCM [26], Group Trust (GT) [22], Eigen Trust (ET) [40], and Reliable IoT Edge Computing (RIEC) [28] trust models.

5.2 Effect of Weight

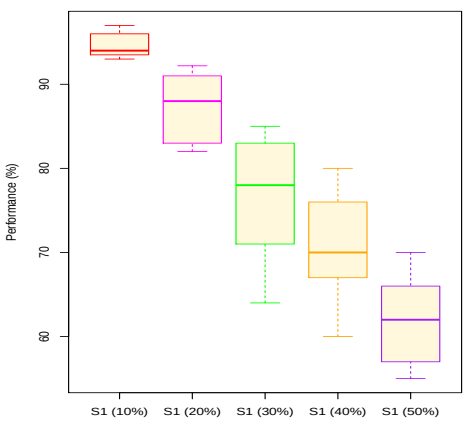
There are same weight wt settings for the end-users. The experimental scenarios for ordinary end users is set at 70%, and the experimental scenario for *scenario-3* end users are set at 30%. The 100 percent PCD is set in this scenario. The experimental results depicted the influence of the weight value. The value of wt greatly dominate the effective service acquisition rate. When $wt = 0$ is provided for effective service, the worst performance is received from the system. Hence, leading to a 90 percent decrease in effective service ratio. When $wt = 1$, the service rate of acquisition is 93.5%. When $wt = 0.85$, the system performs optimally, and the end-user receives effective service with a 95.37% probability.

5.3 Results and Performance Analysis

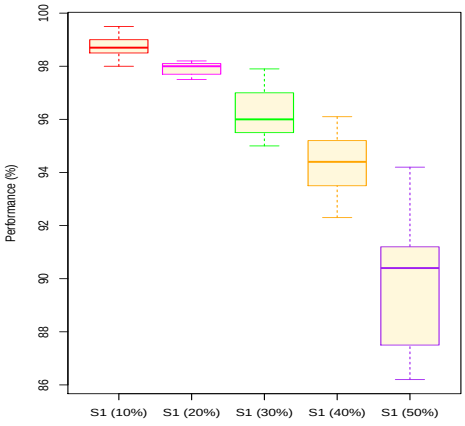
5.3.1 Solitary Attacks

The main goal for studying model performance in service assurance quality is (a) a number of attack models ranging from scenario 1 to 3, (b) The end-user ratio between 10 to 50 percent, and (c) The cooperation degree from 10 to 100 percent. In addition, the availability of effective services is measured in the computing system of edge-of-things for different experimental configurations.

The collaboration degrees of end-users are set to be 10 and 100 percent. The first is the smaller number of Edge-of-Things users in smart cities, and the second is the large number of devices in EoT locations in smart cities.It is observed that the gathering of information on trust evaluation helps in evaluating end-user credibility accurately. With time, the system service acquisition effectively improves. Hence, there is performance improvement. In 10, 20, 30, 40, and 50 percent of malicious end-users scenarios, the proposed model achieved 99.5%, 98.1%, 97.9%, 96.1%, and 93.7% effective service access rates respectively in the 5000 rounds of interaction. The experiment results illustrate the

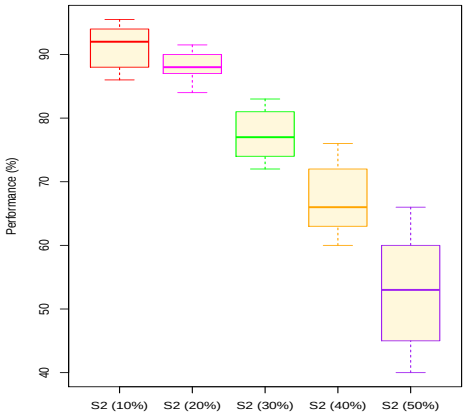


(a) Scenario 1 with 10% PCD

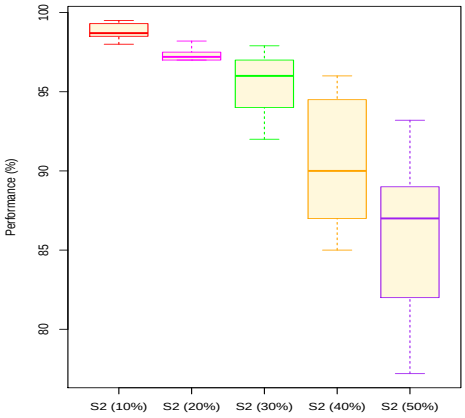


(b) Scenario 1 with 100% PCD

Fig. 2: Performance of proposed RTM scheme against good and bad mouthing attacks

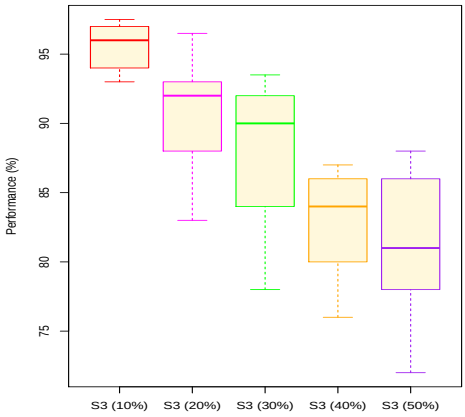


(a) Scenario 2 with 10% PCD

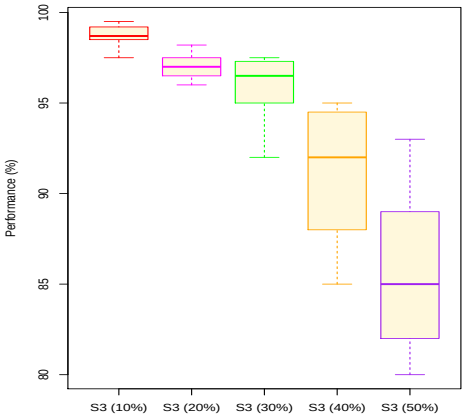


(b) Scenario 2 with 100% PCD

Fig. 3: Performance of proposed RTM scheme against ballot-stuffing attacks

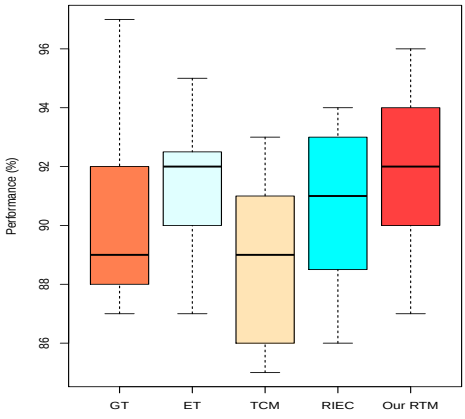


(a) Scenario 3 with 10% PCD

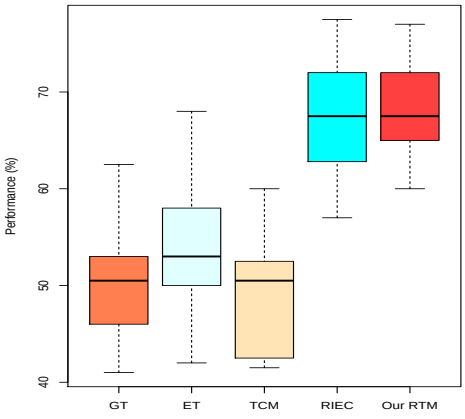


(b) Scenario 3 with 100% PCD

Fig. 4: Performance of proposed RTM scheme against selective-behaviour attacks

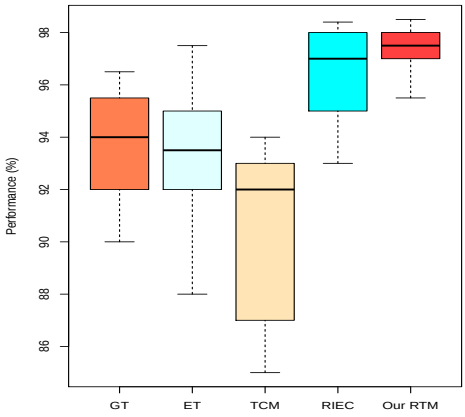


(a) Scenario 2 with 10% malicious nodes

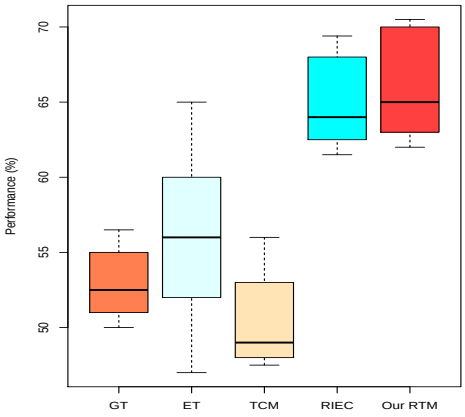


(b) Scenario 2 with 50% malicious nodes

Fig. 5: Performance comparison of proposed RTM with different trust models (PCD 10%)

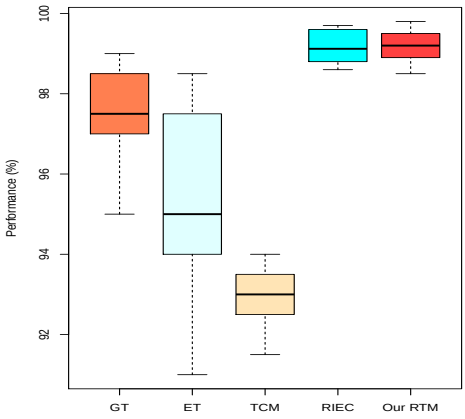


(a) Scenario 2 with 10% malicious nodes

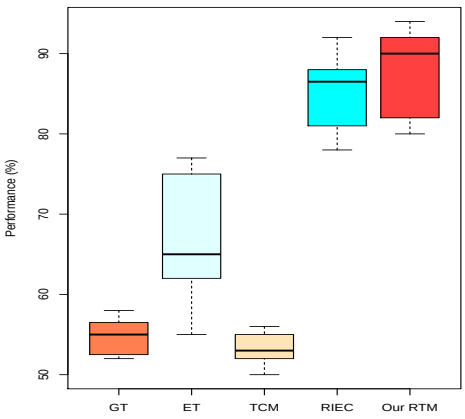


(b) Scenario 2 with 50% malicious nodes

Fig. 6: Performance comparison of proposed RTM with different trust models (PCD 50%)



(a) Scenario 2 with 10% malicious nodes



(b) Scenario 2 with 50% malicious nodes

Fig. 7: Performance comparison of proposed RTM with different trust models (PCD 100%)

effective performance of the proposed model in ensuring the quality of service and quality of protection of the edge of things computing system.

The performance of *scenario-1*, *scenario-2* and *scenario-3* is shown in Figure 2, Figure 3 and Figure 4 respectively. Similarly, as in Figure 2, model performance is improved with time. Under *scenario-2* and *scenario-3*, end-user distribution is 50% malicious. It leads to an effective service acquisition rate in 5000 interaction rounds.

As there is limited feedback information in edge computing with 10 percent idle, the end-user can not estimate credibility. The system performance will vary over time in Figure 2a, Figure 3a and Figure 4a. In the edge-cloud ecosystem, if EoT with 30% and 50% systems, model performance lies between 10% and 100%.

5.3.2 Complex Attacks

In the area of EoT, various malicious users exist at a single time. So, for studying the robustness and effectiveness of the trust model, complex attacks are considered based on performance in experiments. In a complex attack scenario, the experimental settings are as follows: (a) the malicious user distribution ratio is 10% to 50%; (b) the PCD values are 10%, 50%, and 100%; and (c) the complex attack model has an effect on task ratio. The 4 cases are being considered.

In case one, scenarios 1 to 3 are randomly selected by end-users with a probability of 1/3. In case two, scenarios 1 to 3 are selected by the end-users with 50, 30, and 10 percent probability. In case three, scenarios 1 to 3 are selected by end-users with a probability of 60, 25, and 15 percent. In case 4, scenario 1 to 3 is selected by end-users with the probability of 50, 30, and 10 percent. The results are depicted in Figure 4. The different findings are with an increase in malicious end users, and the task success ratio decreases. The rate of decrease is slow, hence predicting the robust performance of the trust model. As the PCD value increases, the task success ratio increases by 50% as malicious users. The complex attack models have varying proportions. The trust model can achieve similar results.

5.3.3 Comparative Experiment

The model performance compared with Trust Computing Mechanism (TCM) [26], Group Trust (GT) [22], and Eigen Trust (ET) [40] and Reliable IoT Edge Computing (RIEC) [28] trust models. TCM was preferred because it is one of the latest IoT systems based on information entropy theory, and RIEC is the latest trust model for IoT edge computing. The direct evaluation information was applied to calculate indirect trust value. In experimental results, the best performance given by the proposed RTM model is is 96% with 10% PCD value and 10% malicious nodes in *scenario-2* as shown in Figure 5a. Similarly, the end-users achieved the 77% accuracy with 10% PCD value and 50% malicious nodes as shown in Figure 5b. The proposed model performed better when there were fewer malicious nodes, while the RIEC model is superior with a large number of malicious nodes with 10% PCD.

The end-user ratio setting is made at 50% malicious, which leads to the worst performance of the GT model that may be due to the problem of randomness caused by less availability of performance evaluation. As the malicious end-user ratio increases, there is a decrease in the performance of the system. In the experimental evaluation, the PCD value set as 50% in *scenario-2* is shown in Figure 6a and Figure 6b. This results in elevation of interaction between the end-users. It is found that the proposed scenario achieves the best performance under 10 to 50 percent of malicious distribution ratios are 98.5% and 70.5%. The TCM model's performance is low in this scenario. The performance of the RIEC trust model is better than existing trust models and very close to the proposed RTM trust model for fewer malicious nodes. However, the proposed RTM model outperforms with an increase in malicious nodes with 50% PCD.

As with limited trust evaluation information, the credibility of the end-user can not be evaluated accurately. In comparison to the PCD with 10 percent, trust evaluation information shows an absolute increase. The rate of task completion is improved. With the value of PCD as 50 percent, it improves the success rate of tasks. When the PCD value is 100 percent in attack *scenario-2*, the edge-of-computing system is extremely busy. As depicted in Figure 7a and Figure 7b, it is found that the results achieved improved a bit with a PCD value of 50 percent and also there is an improvement in the success rate of the system. The 99.8% performance is achieved with the proposed RTM trust model. The comparative results are depicted in Figure 5, Figure 6 and Figure 7. The existing RIEC model is superior to the existing model and also gives a high performance as proposed RTM in some cases. However, the proposed RTM model gives better performance with a different number of malicious nodes with varying PCD.

6 Conclusion

This paper presented a robust trust management scheme for EoT in smart cities. The proposed framework maintains trust with the quality of protection (QoP) and quality of service (QoS). The service quality is improved by selecting trusted participants with a collaboration filtration method. The proposed recommendation system is capable of calculating the trust score with no interaction history. The extensive experiment is conducted with various types of attacks with a different number of malicious nodes. The proposed RTM trust provisioning scheme achieved the accuracy up to 99.5% for solitary attacks, and 99.8% for complex attacks with 10% malicious nodes. The results demonstrate that the proposed system works better in contrast with existing trust management schemes. The association of multiple smart devices is possible with QoP with the desired secure service level agreement (SSLA). The proposed RTM model is most suitable for broad infrastructure in a smart city. In the future, this technique could be extended with smart contracts between reliable edge nodes. A new technique is also required to handle a large number of malicious edge nodes.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[3] K. Gai, M. Qiu, Z. Xiong, and M. Liu, "Privacy-preserving multi-channel communication in edge-of-things," *Future Generation Computer Systems*, vol. 85, pp. 190–200, 2018.

[4] P. Singh, A. Kaur, G. S. Aujla, R. S. Batth, and S. Kanhere, "Daas: Dew computing as a service for intelligent intrusion detection in edge-of-things ecosystem," *IEEE Internet of Things Journal*, 2020.

[5] I. Bäumler and H. Kotzab, "Scenario-based development of intelligent transportation systems for road freight transport in germany," in *Urban Freight Transportation Systems*. Elsevier, 2020, pp. 183–202.

[6] M. A. Salem, S. M. A. El-Kader, M. I. Youssef, and I. F. Tarrad, "M2m in 5g communication networks: Characteristics, applications, taxonomy, technologies, and future challenges," in *Fundamental and Supportive Technologies for 5G Mobile Networks*. IGI Global, 2020, pp. 309–321.

[7] M. Foth, L. Forlano, and M. Bilandzic, "Mapping new work practices in the smart city," in *Handbuch Soziale Praktiken und Digitale Alltagswelten*. Springer, 2020, pp. 169–181.

[8] P. T. Lam and W. Yang, "Factors influencing the consideration of public-private partnerships (ppp) for smart city projects: Evidence from hong kong," *Cities*, vol. 99, p. 102606, 2020.

[9] G. S. Aujla, A. Singh, M. Singh, S. Sharma, N. Kumar, and K.-K. R. Choo, "Blocked: Blockchain-based secure data processing framework in edge envisioned v2x environment," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5850–5863, 2020.

[10] G. S. Aujla, N. Kumar, M. Singh, and A. Y. Zomaya, "Energy trading with dynamic pricing for electric vehicles in a smart city environment," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 169–183, 2019.

[11] S. Misra, S. Das, M. Khatua, and M. S. Obaidat, "Qos-guaranteed bandwidth shifting and redistribution in mobile cloud environment," *IEEE Transactions on Cloud Computing*, vol. 2, no. 2, pp. 181–193, 2013.

[12] S. Misra and N. Saha, "Detour: Dynamic task offloading in software-defined fog for iot applications," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1159–1166, 2019.

[13] T.-H. Woo and K.-B. Jang, "Cloud computing based analysis incorporated with the internet of things (iot) in nuclear safety assessment for fukushima dai-ichi disaster," *Journal of The Korea Internet of Things Society*, vol. 6, no. 1, pp. 73–81, 2020.

[14] S. Bera, S. Misra, S. K. Roy, and M. S. Obaidat, "Soft-wsn: Software-defined wsn management system for iot applications," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2074–2081, 2016.

[15] H. Cao, S. Wu, G. S. Aujla, Q. Wang, L. Yang, and H. Zhu, "Dynamic embedding and quality of service-driven adjustment for cloud networks," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1406–1416, 2019.

[16] G. S. Aujla, A. Jindal, and N. Kumar, "Evaas: Electric vehicle-as-a-service for energy trading in sdn-enabled smart transportation system," *Computer Networks*, vol. 143, pp. 247–262, 2018.

[17] G. S. Aujla, N. Kumar, A. Y. Zomaya, and R. Ranjan, "Optimal decision making for big data processing at edge-cloud environment: An sdn perspective," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 778–789, 2017.

[18] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Future generation computer systems*, vol. 56, pp. 684–700, 2016.

[19] D. Gessner, A. Olivereau, A. S. Segura, and A. Serbanati, "Trustworthy infrastructure services for a secure and privacy-respecting internet of things," in *2012 IEEE 11th international conference on trust, security and privacy in computing and communications*. IEEE, 2012, pp. 998–1003.

[20] C. Wang, R. S. Batth, P. Zhang, G. S. Aujla, Y. Duan, and L. Ren, "Vne solution for network differentiated qos and security requirements: from the perspective of deep reinforcement learning," *Computing*, pp. 1–23, 2021.

[21] Z. Su, L. Liu, M. Li, X. Fan, and Y. Zhou, "Servicetrust: Trust management in service provision networks," in *2013 IEEE International Conference on Services Computing*. IEEE, 2013, pp. 272–279.

[22] X. Fan, L. Liu, M. Li, and Z. Su, "Grouptrust: dependable trust management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1076–1090, 2016.

[23] H. Hui, C. Zhou, X. An, and F. Lin, "A new resource allocation mechanism for security of mobile edge computing system," *IEEE Access*, vol. 7, pp. 116 886–116 899, 2019.

[24] I. U. Din, M. Guizani, B.-S. Kim, S. Hassan, and M. K. Khan, "Trust management techniques for the internet of things: A survey," *IEEE Access*, vol. 7, pp. 29 763–29 787, 2018.

[25] M. Li, Q. Guan, X. Jin, C. Guo, X. Tan, and Y. Gao, "Personalized pre-trust reputation management in social p2p network," in *2016 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2016, pp. 1–5.

[26] J. Yuan and X. Li, "A reliable and lightweight trust computing mechanism for iot edge devices based on multi-source feedback information fusion," *Ieee Access*, vol. 6, pp. 23 626–23 638, 2018.

[27] M. Bahutair, A. Bouguettaya, and A. G. Neiat, "Multi-perspective trust management framework for crowdsourced iot services," *IEEE Transactions on Services Computing*, 2021.

[28] B. Wang, M. Li, X. Jin, and C. Guo, "A reliable iot edge computing trust management mechanism for smart cities," *IEEE Access*, vol. 8, pp. 46 373–46 399, 2020.

[29] S. Aalibagi, H. Mahyar, A. Movaghar, and H. E. Stanley, "A matrix factorization model for hellinger-based trust management in social internet of things," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[30] Z.-J. Liu, S. Chernov, and A. V. Mikhaylova, "Trust management and benefits of vehicular social networking: An approach to verification and safety," *Technological Forecasting and Social Change*, vol. 166, p. 120613, 2021.

[31] R. Chen, J. Guo, and F. Bao, "Trust management for soa-based iot and its application to service composition," *IEEE Transactions on Services Computing*, vol. 9, no. 3, pp. 482–495, 2014.

[32] Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for internet of things," *Journal of network and computer applications*, vol. 42, pp. 120–134, 2014.

[33] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "Securing fog computing for internet of things applications: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601–628, 2017.

[34] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE internet of things journal*, vol. 3, no. 6, pp. 1171–1181, 2016.

[35] T. He, E. N. Ciftcioglu, S. Wang, and K. S. Chan, "Location privacy in mobile edge clouds: A chaff-based approach," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2625–2636, 2017.

[36] C. Pahl, N. El Ioini, S. Helmer, and B. Lee, "An architecture pattern for trusted orchestration in iot edge clouds," in *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 2018, pp. 63–70.

[37] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of internet of things," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 46–59, 2015.

[38] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.

[39] C. Vallati, A. Virdis, E. Mingozzi, and G. Stea, "Exploiting lte d2d communications in m2m fog platforms: Deployment and practical issues," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*. IEEE, 2015, pp. 585–590.

[40] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of the 12th international conference on World Wide Web*, 2003, pp. 640–651.